

**IN THE CLAIMS:**

- 1 | 1. (Currently Amended) A method for comparing a first ~~order-independent data-set~~direc-  
2 | tory comprising unique elements with a second directory ~~order-independent data-set~~ com-  
3 | prising unique elements, comprising:
- 4 |       (a) for each entry in the first ~~data-set~~directory, placing the entry in a hash table,  
5 | wherein the first directory~~data-set~~ is stored on a source storage system;
- 6 |       (b) selecting an entry from the second directory~~data-set~~, wherein the second direc-  
7 | tory~~data-set~~ is located on a destination storage system and the source storage system and  
8 | the destination storage system are separate stand alone storage systems;
- 9 |       (c) looking up the selected entry in the hash table;
- 10 |       (d) removing, in response to locating the selected entry in the hash table, the se-  
11 | lected entry from the hash table;
- 12 |       (e) determining if additional second directory~~data-set~~ entries exist;
- 13 |       (f) looping to step (b) in response to identifying additional second directory~~data~~  
14 | ~~set~~ entries; and
- 15 |       (g) reporting a difference between the first directory~~data-set~~ and the second direc-  
16 | tory~~data-set~~ in response to at least one first directory~~data-set~~ entry remaining in the hash  
17 | table.
- 1 | 2. (Currently Amended) The method of claim 1 further comprising identifying, in re-  
2 | sponse to not locating the selected entry in the hash table, that the selected entry is second  
3 | directory~~data-set~~ unique.
- 1 | 3. (Previously Presented) The method of claim 1 further comprising performing, in re-  
2 | sponse to not locating the selected entry in the hash table, a remedial function.

- 1 4. (Currently Amended) The method of claim 3 wherein the remedial function comprises  
2 | deleting the selected entry of the second directory~~data-set~~.
- 1 5. (Currently Amended) The method of claim 1 further comprising identifying in re-  
2 sponse to no additional entries existing, any remaining entries in the hash table data as  
3 | being first directory~~data-set~~ unique.
- 1 6. (Previously Presented) The method of claim 1 further comprising performing in re-  
2 sponse to no additional entries existing, a remedial function.
- 1 7. (Currently Amended) The method of claim 6 wherein the remedial function comprises  
2 | deleting the selected entry of the first directory~~data-set~~.
- 1 8. (Currently Amended) The method of claim 6 wherein the remedial function comprises  
2 | transferring the selected entry from the first directory~~data-set~~ to the second directory~~data~~  
3 | ~~set~~.
- 1 9. (Currently Amended) The method of claim 1 wherein the step of removing the selected  
2 entry from the hash table occurs in response to identifying a match between a selected  
3 | entry of the first directory~~data-set~~ and an entry of the second directory~~data-set~~.
- 1 10. (Original) The method of claim 1 wherein the hash table comprises a B-tree.
- 1 11. (Original) The method of claim 1 wherein the hash table comprises a fast lookup data  
2 structure.
- 1 12. – 16. (Cancelled)

1 17. (Previously Presented) A system for comparing a first data set with a second data set,  
2 the system comprising:

3 (a) means for placing each entry of the first data set in a hash table, wherein the  
4 first data set is stored on a source storage system;

5 (b) means for selecting an entry from the second data set, wherein the second data  
6 set is located on a destination storage system and the source storage system and the desti-  
7 nation storage system are separate stand alone storage systems;

8 (c) means for looking up the selected entry in the hash table;

9 (d) means for removing, in response to locating the selected entry in the hash ta-  
10 ble, the selected entry from the hash table;

11 (e) means for determining if additional second data set entries exist;

12 (f) means for looping to step (b) in response to identifying additional second data  
13 set entries; and

14 (g) means for reporting a difference between the first data set and the second data  
15 set in response to at least one first data set entry remaining in the hash table.

1 18. (Original) The system of claim 17 wherein the hash table comprises a B-tree.

1 19. (Currently Amended) A computer readable medium, ~~including containing executable~~  
2 ~~program instructions executing on a computer, the program instructions including in-~~  
3 ~~structions for performing the steps of~~ executed by a processor, comprising:

4 (a) for each entry in a first data set, program instructions that ~~placing place~~ the  
5 entry in a hash table, wherein the first data set is stored on a source storage system;

6 (b) program instructions that ~~selecting select~~ an entry from a second data set,  
7 wherein the second data set is located on a destination storage system and the source  
8 storage system and the destination storage system are separate stand alone storage sys-  
9 tems;

10 (c) program instructions that ~~looking look~~ up the selected entry in the hash table;

- 11 |       (d) program instructions that removing~~remove~~, in response to locating the se-
- 12 | lected entry in the hash table, the selected entry from the hash table;
- 13 |       (e) program instructions that determining~~determine~~ if additional second data set
- 14 | entries exist;
- 15 |       (f) program instructions that looping~~loop~~ to step (b) in response to identifying
- 16 | additional second data set entries; and
- 17 |       (g) program instructions that reporting~~report~~ a difference between the first data
- 18 | set and the second data set in response to at least one first data set entry remaining in the
- 19 | hash table.

1   20. (Previously Presented) A method for comparing a first data set with a second data set,

2   comprising:

3       creating a hash table of entries of the first data set, wherein the first data set is

4   stored on a source storage system;

5       locating, for each entry in the second data set, an entry in the hash table, wherein

6   the second data set is located on a destination storage system and the source storage sys-

7   tem and the destination storage system are separate stand alone storage systems;

8       removing, in response to locating an entry in the hash table, the located entry; and

9       recording, in response to at least one entry remaining in the hash table, a differ-

10 | ence between the first data set and the second data set.

1   21. (Previously Presented) A method for comparing a first data set with a second data set,

2   comprising:

3       creating a hash table of entries of the first data set, wherein the first data set is

4   stored on a source storage system;

5       locating, for each entry in the second data set, an entry in the hash table, wherein

6   the second data set is located on a destination storage system and the source storage sys-

7   tem and the destination storage system are separate stand alone storage systems;

8       removing, in response to locating an entry in the hash table, the located entry;

9           recording, in response to not locating an entry in the hash table, that the entry in  
10 the second data set is second date set unique; and  
11           reporting a difference between the first data set and the second data set in re-  
12 sponse to at least one first data set entry remaining in the hash table.

1   22. (Previously Presented) A method for comparing a first data set with a second data set,  
2 comprising:

3           (a) selecting an entry from the first data set, wherein the first data set is stored on  
4 a source storage system;

5           (b) determining if the selected entry from the first data set is in a hash table;

6           (c) adding, in response to determining that the selected entry from the first data  
7 set is not in the hash table, the selected entry from the first data set to the hash table;

8           (d) removing from the hash table, in response to determining that the selected en-  
9 try from the first data set is in the hash table, the selected entry from the first data set;

10          (e) selecting an entry from the second data set, wherein the second data set is lo-  
11 cated on a destination storage system and the source storage system and the destination  
12 storage system are separate stand alone storage systems;

13          (f) determining if the selected entry from the second data set is in the hash table;

14          (g) adding, in response to determining that the selected entry from the second data  
15 set is not in the hash table, the selected entry from the second data set to the hash table;

16          (h) removing, in response to determining that the selected entry from the second  
17 data set is in the hash table, the selected entry from the second data set from the hash ta-  
18 ble;

19          (i) independently continuing steps (a) through (d) and (e) through (h) for all en-  
20 tries in the first and second data sets until both the first and second data sets have been  
21 completely processed; and

22          (j) reporting a difference between the first data set and the second data set in re-  
23 sponse to at least one entry remaining in the hash table.

1 23. (Previously Presented) The method of claim 22 wherein the step of adding the se-  
2 lected entry from the first data set to the hash table further comprises including informa-  
3 tion with the selected entry from the first data set identifying the selected entry from the  
4 first data set as originating from the first data set.

1 24. (Previously Presented) The method of claim 22 wherein the step of adding the se-  
2 lected entry from the second data set to the hash table further comprises including infor-  
3 mation with the selected entry from the second data set identifying the selected entry  
4 from the second data set as originating from the second data set.

1 25. (Previously Presented) The method of claim 22 wherein the step of removing the se-  
2 lected entry from the second data set from the hash table occurs in response to identifying  
3 a match between a selected entry from the second data set and an entry from the first data  
4 set.

1 26. (Previously Presented) The method of claim 22 further comprising:  
2 (k) recording all entries remaining in the hash table as being unique to either the  
3 first data set or the second data set.

1 27. (Original) The method of claim 22 wherein the hash table comprises a B-tree.

1 28. (Original) The method of claim 22 wherein the hash table comprises a fast lookup  
2 data structure.

1 29. (Original) The method of claim 22 wherein the first data set comprises a set of direc-  
2 tory entries on a source system.

1 30. (Original) The method of claim 22 wherein the second data set comprises a set of di-  
2 rectory entries on a destination system.

1 31. (Original) The method of claim 22 wherein the first data set and second data set are  
2 on different storage devices.

1 32. (Currently Amended) A system for performing a consistency check of a source direc-  
2 tory replicated to a destination directory by comparing entries in the source and destina-  
3 tion directories, the system comprising:

4 one or more storage disks configured to store one or more of a group consisting of  
5 the source directory and the destination directory; ~~and~~

6 a process configured to compare entries in the source directory with entries in the  
7 destination directory by walking the source and destination directories only once,  
8 whereby utilization of storage subsystems associated with the source and destination di-  
9 rectories is limited by only walking each of the source and destination directories once,  
10 and further configured to report a difference between the source directory and the desti-  
11 nation directory, wherein the source directory is located on a source storage system and  
12 the destination directory is located on a destination storage system and the source storage  
13 system and the destination storage system are separate stand alone storage systems; and  
14 the process is further configured to remove matching entries from a hash table,  
15 whereby future look up operations in the hash table are enabled to be performed faster  
16 due to a smaller size of the hash table.

1 33. (Original) The system of claim 32 wherein the process executes on a computer asso-  
2 ciated with the source directory.

1 34. (Original) The system of claim 32 wherein the process executes on a computer asso-  
2 ciated with the destination directory.

1 35. (Cancelled)

1 36. (Currently Amended) A system for performing a consistency check of a source direc-  
2 tory and a destination directory by comparing entries in the source and destination direc-  
3 tories, the system comprising:

4       the source directory stored on a source storage system;  
5       the destination directory stored on a destination storage system, wherein the  
6 source storage system and the destination storage system are separate stand alone storage  
7 systems; and

8       a processor configured to select alternating entries from the source and destination  
9 directories to be added to a hash table and further adapted to remove matching entries  
10 from the hash table, whereby a size of the hash table is limited to a number of dissimilar  
11 entries of the source and destination directories, and further configured to report a differ-  
12 ence between the source directory and the destination directory in response to the number  
13 of dissimilar entries being greater than zero, ~~wherein the source directory is located on a~~  
14 ~~source storage system and the destination directory is located on a destination storage~~  
15 ~~system and the source storage system and the destination storage system are separate~~  
16 ~~stand alone storage systems.~~

1 37. (Currently Amended) A system for comparing entries in a source directory with en-  
2 tries on a destination directory to ensure consistency of replicated data between the  
3 source and destination directories, the system comprising:

4       the source directory stored on a source storage system;  
5       the destination directory stored on a destination storage system, wherein the  
6 source storage system and the destination storage system are separate stand alone storage  
7 systems; and

8       a computer associated with at least one of the source and destination directories,  
9 the computer comprising a directory comparison process configured to perform a com-  
10 parison of entries in the source and destination directories by walking each directory once  
11 and placing entries in a hash table and further configured to remove matching entries  
12 from the hash table, whereby computational cost is reduced for future look up operations



13 | in the hash table, ~~wherein the source directory is located on a source storage system and~~  
14 | ~~the destination directory is located on a destination storage system and the source storage~~  
15 | ~~system and the destination storage system are separate stand alone storage systems.~~

1 38. (Previously Presented) The system of claim 37 wherein the directory comparison  
2 process is further configured to alternate in selecting entries from the source and destina-  
3 tion directories when walking the source and destination directories.

1 39. (Previously Presented) The method of claim 1 wherein the step of reporting com-  
2 prises recording the difference on a disk.

1 40. (Previously Presented) The method of claim 22 wherein the step of reporting com-  
2 prises recording the difference on a disk.

1 41. (Previously Presented) The system of claim 32 wherein the process is further adapted  
2 to report the difference by recording the difference on the storage disks.